

## INTERPOLASI SPLINE DALAM MATLAB

Oleh :  
Bambang Suharjo  
Fakultas Teknik Universitas Muhammadiyah Gresik

### Abstrak

Fitting data dengan metode least square, interpolasi linear, interpolasi lagrange maupun dengan fungsi besel mengandung kelemahan masing-masing. Model spline dapat dijadikan salah satu alternatif untuk menanggulangi kelemahan tersebut. Salah satu model spline yang banyak digunakan adalah spline kuadrat. Spline kuadrat dapat disusun sebagai

$$S_i(x) = y_i + z_i(x - x_i) + \frac{z_{i+1} - z_i}{2(x_{i+1} - x_i)}(x - x_i)^2$$

Koefisiennya dapat ditemukan dengan memilih  $z_0$  dan kemudian menggunakan hubungan recurrent :

$$z_{i+1} = -z_i + 2 \frac{y_{i+1} - y_i}{x_{i+1} - x_i}$$

Menggunakan interpolasi spline, dapat disusun suatu program Matlab yang disesuaikan dari Collier dan Kaw (2006), Spline Toolbox (2006), Spline Interpolation (2006) dan Spline Interpolation in Matlab (2006) untuk menghasilkan grafik dan persamaan spline dari setiap pasang koordinat ke pasangan koordinat berikutnya.

Kata kunci : interpolasi, spline, Matlab

### 1. Pendahuluan

Fitting data pada umumnya dapat dilakukan dengan metode kuadrat terkecil (*least squares fitting method*). Metode ini pada umumnya dapat dilakukan dengan mudah bila bentuk dari fungsi kurvanya diketahui dan sederhana, tetapi bilamana bentuk dari fungsi kurvanya sendiri belum diketahui, maka metode pendekatan atau aproksimasi dapat dilakukan dengan berbagai cara, misalnya dengan menguraikan fungsinya dalam deret, misalnya deret Taylor bilamana fungsi dan turunannya pada titik-titik tertentu diketahui, atau menggunakan interpolasi Lagrange, bilamana hanya fungsinya saja yang diketahui pada beberapa titik. Metode lain adalah menguraikan fungsinya dalam fungsi-fungsi yang sifatnya telah diketahui, misalnya fungsi Bessel atau polinomial Hermit, dan sebagainya [1]. Metode-metode tersebut di atas pada umumnya dapat berhasil dengan baik, tetapi ada kalanya dapat pula mengalami kegagalan, terutama untuk kedua titik data yang berada di ujung data, sangat sukar untuk menentukan turunannya. Dalam tulisan ini akan dibahas interpolasi spline yang sangat berguna untuk fitting data-data yang bentuk fungsinya maupun turunannya tidak diketahui (Lasijo, 2001).

## 2. Interpolasi Linear

Interpolasi linear adalah interpolasi yang diperoleh dengan cara menghubungkan dua titik yang mengapit daerah yang akan dicari interpolasinya. Secara aljabar, untuk setiap  $S_i$  sebagai fungsi linear yang dibangun sebagai berikut (Spline Interpolation, 2006).

$$S_i(x) = y_i + \frac{y_{i+1} - y_i}{x_{i+1} - x_i}(x - x_i) \quad \dots\dots\dots (1)$$

Garis harus kontinu pada setiap titik, yaitu :

$$S_i(x_i) = S_{i+1}(x_i) \quad , \quad i = 0, \dots, k - 1$$

Sehingga secara mudah dapat dilihat bahwa :

$$S_{i-1}(x_i) = y_{i-1} + \frac{y_i - y_{i-1}}{x_i - x_{i-1}}(x_i - x_{i-1}) = y_i \quad \dots\dots\dots (2)$$

$$S_i(x_i) = y_i + \frac{y_{i+1} - y_i}{x_{i+1} - x_i}(x_i - x_i) = y_i$$

## 3. Interpolasi Spline

Spline merupakan suatu kurva yang dibangun dari potongan-potongan polynomial (piecewise polynomial) dengan titik-titik belok disebut knot (spline polynomial, 2006). Diberikan  $n+1$  knot  $x_i$  dengan

$$x_0 < x_1 < \dots < x_{n-1} < x_n,$$

Dengan  $n+1$  nilai-nilai knot  $y_i$  dicoba untuk menemukan suatu fungsi *spline* derajat  $n$  , yaitu :

$$S(x) := \begin{cases} S_0(x) & x \in [x_0, x_1] \\ S_1(x) & x \in [x_1, x_2] \\ \vdots & \vdots \\ S_{n-1}(x) & x \in [x_{n-1}, x_n] \end{cases} \quad \dots\dots\dots (3)$$

Dengan setiap  $S_i(x)$  adalah polinomial derajat  $k$ .

Menggunakan interpolasi polinomial, polinomial derajat  $n$  yang menginterpolasi himpunan data adalah secara unik didefinisikan dengan titik-titik data. Spline derajat  $n$  yang menginterpolasi himpunan data yang sama tidak

secara unik didefinisikan, dan kita mengisi dalam  $n-1$  derajat bebas tambahan untuk menyusun suatu interpolan yang unik.

Salah satu model spline yang banyak digunakan adalah spline kuadrat. Spline kuadrat dapat disusun sebagai :

$$S_i(x) = y_i + z_i(x - x_i) + \frac{z_{i+1} - z_i}{2(x_{i+1} - x_i)}(x - x_i)^2 \dots\dots\dots (4)$$

Koefisiennya dapat ditemukan dengan memilih  $z_0$  dan kemudian menggunakan hubungan recurrent :

$$z_{i+1} = -z_i + 2 \frac{y_{i+1} - y_i}{x_{i+1} - x_i} \dots\dots\dots (5)$$

Selanjutnya dimasukkan dalam persamaan (4) untuk menghasilkan persamaan spline yang dimaksud.

#### 4. Implementasi dengan Matlab

Kesulitan utama yang muncul dari proses interpolasi (terutama interpolasi spline) adalah teknis komputasi. Oleh karena itu perlu suatu mekanisme pendukung. Software Matlab dapat digunakan untuk mempermudah pelaksanaan perhitungan interpolasi, bahkan sampai dengan penyusunan fungsi dan penggambaran grafiknya.

##### 4.1 Interpolasi Linear

Misalnya akan dicari interpolasi dari titik-titik sebagai berikut.

$$\{(80,6.47), (0,6.0), (-60,5.58), (-160,4.72), (-260,3.58), (-340,2.45)\}$$

Menggunakan interpolasi linear, dapat disusun suatu program Matlab yang disesuaikan dari Collier dan Kaw (2006), Spline Toolbox (2006), Spline Interpolation (2006) dan Spline Interpolation in Matlab (2006) untuk menghasilkan grafik dan persamaan linear dari setiap pasang koordinat ke pasang koordinat berikutnya.

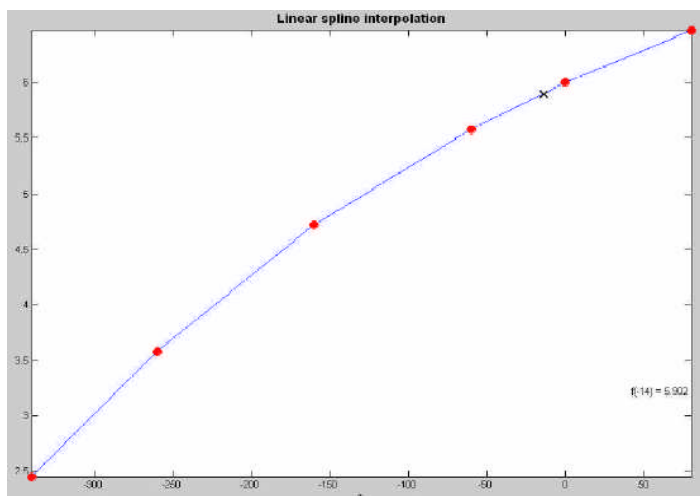
```
clear all
% INPUTS: Enter the following
% Array of x-data
x=[80 0 -60 -160 -260 -340];
% Array of y-data
y=[6.47 6.0 5.58 4.72 3.58 2.45];
% Value of x at which y is desired
xdesired = -14;
% SOLUTION
% This calculates window size to be used in figures
set(0,'Units','pixels')
scnsize = get(0,'ScreenSize');
wid = round(scnsize(3));
hei = round(0.95*scnsize(4));
wind = [1, 1, wid, hei];
% The following functions sort the x and y arrays in ascending x order.
xy=cat(1,x,y);
xy=xy';
xy=sortrows(xy,1);
for i = 1:numel(x)
```

```

x(i)=xy(i,1);
y(i)=xy(i,2);
end
%%%%%%%%% LINEAR INTERPOLATION %%%%%%%%%%
figure('Position',wind)
ymax=max(y);
ymin=min(y);
z=sym('z');
title('Linear spline interpolation','Fontweight','bold','FontSize',14)
text(0,1,'Sorted Data','Fontweight','bold')
midp=(0.95+(0.98-0.03*numel(x)))/2;
text(0,midp,'x =')
text(0.13,midp,y =)
for i = 1:numel(x)
text(0.05,0.98-0.03*i,num2str(x(i)))
text(0.18,0.98-0.03*i,num2str(y(i)))
end
t=0.9-0.02*numel(x)-0.06;
text(0,t,'Equations of the spline','Fontweight','bold')
for i = 1:numel(x)-1
m=(y(i+1)-y(i))/(x(i+1)-x(i));
f=m*(z-x(i))+y(i);
text(0,t-0.04*i-0.01,['f(x)_' ,num2str(i),' = ' ,num2str(m),'x + ' ,num2str(-m*x(i)+y(i))])
end
axis off
figure('Position',wind)
for i = 1:numel(x)-1
m=(y(i+1)-y(i))/(x(i+1)-x(i));
z=sym('z');
f=m*(z-x(i))+y(i);
if xdesired < x(i+1) & xdesired >= x(i)
fxldesired=subs(f,z,xdesired);
end
ezplot(f,[x(i),x(i+1)])
hold on
end
plot(x,y,'ro','MarkerSize',10,'MarkerFaceColor',[1,0,0])
plot(xdesired,fxldesired,'kx','Linewidth',2,'MarkerSize',12)
title('Linear spline interpolation','Fontweight','bold','FontSize',14)
text(0.1*(max(x)-min(x)),0.8*(max(y)-min(y)),['f(',num2str(xdesired),') = ' ,num2str(fxldesired)])
xlim([min(x) max(x)])
ylim([min(y) max(y)])

```

Hasilnya adalah sebagai berikut.



Gambar 1. Grafik interpolasi linear dengan Matlab

Persamaan yang dihasilkan dari interpolasi linear di atas menjadi

**LINEAR INTERPOLATION**

$$f(x_1) = 0.014125x + 7.2525$$

$$f(x_2) = 0.0114x + 6.544$$

$$f(x_3) = 0.0086x + 6.096$$

$$f(x_4) = 0.007x + 6$$

$$f(x_5) = 0.005875x + 6$$

Gambar 2. Output persamaan interpolasi linear dengan Matlab

## 4.2 Interpolasi Spline

Selanjutnya, apabila pasangan koordinat di atas diinterpolasikan dengan metode kuadrat spline dengan software Matlab, maka programnya dilanjutkan sebagai berikut.

```

%%%%%%%%%% QUADRATIC SPLINE INTERPOLATION %%%%%%%%%%%
% The following displays the sorted data for the quadratic spline
figure('Position',wind)
title('Quadratic spline interpolation','fontWeight','bold','FontSize',14)
text(0.1,'Sorted Data','fontWeight','bold')
midp=(0.95+(0.98-0.03*numel(x)))/2;
text(0,midp,'x =')
text(0.13,midp,'y =')
for i = 1:numel(x)
text(0.05,0.98-0.03*i,num2str(x(i)))
text(0.18,0.98-0.03*i,num2str(y(i)))
end
axis off
% The following assembles the matrix needed to find the coefficients of the spline
% polynomials.
% Assembly of A
n=numel(x);
A=zeros(3*(n-1));
for i=1:n-1
for j=0:1
for k=0:2
A(2*i-1+j+1,3*i-3+k+1)=x(i+j)^k;
end
end
end
for i=1:n-2
for j=0:1
for k=0:1
A(2*(n-1)+i+1,3*i-2+k+j*3+1)=((-1)^j)*(2*x(i+1))^k;
end
end
end
A(1,3)=1;
% Assembly of Y
Y=zeros(3*(n-1),1);
for i=0:n-2
for j=0:1
Y(2*(i+1)+j)=y(i+j+1);

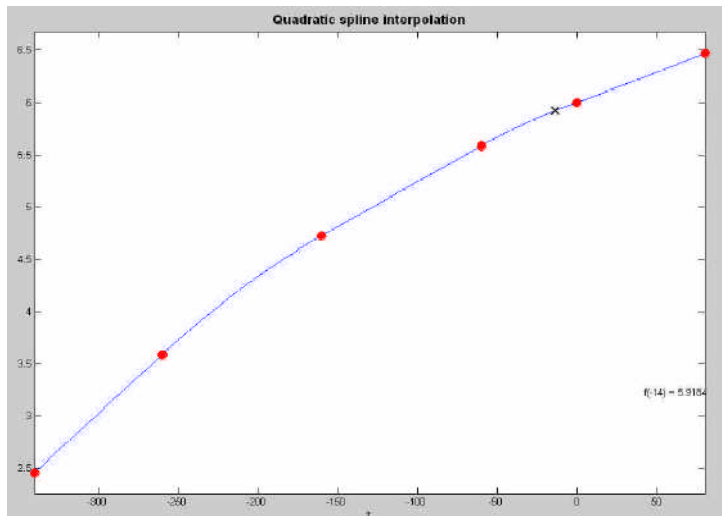
```

```

end
end
% Calculation of C (Coefficient Matrix)
C=inv(A)*Y;
% The following prints the A, Y, and C matrix in the form A^-1*Y=C
figure('Position',wind)
title('Quadratic spline interpolation','Fontweight','bold','FontSize',14)
axis off
hold on
fo='% 1.4g';
sh=1/((3*(n-1))+8);
sv=1/((3*(n-1)));
for c=1:3*(n-1)+8
for r = 1:3*(n-1)
if c <= 3*(n-1)
text(0+(c-1)*sh,1-(r-1)*sv,num2str(A(r,c),fo,'FontSize',10)
end
if c == 3*(n-1)+8
text(0+(c-1)*sh,1-(r-1)*sv,num2str(C(r),fo,'FontSize',10)
end
if c == 3*(n-1)+4
text(0+(c-1)*sh,1-(r-1)*sv,num2str(Y(r),fo,'FontSize',10)
end
end
end
text((3*(n-1))*sh-0.5*sh,1,'^-1','fontSize',12)
text((3*(n-1)+5)*sh,0.5,'=', 'fontSize',12)
text((3*(n-1)+1)*sh,0.5,'X','fontSize',12)
% The following prints the equations of the splines
figure('Position',wind)
title('Quadratic spline interpolation','Fontweight','bold','FontSize',14)
text(0,1,'Equations of the spline','Fontweight','bold')
for i=0:n-2
f=C(3*i+1)+C(3*i+2)*z+C(3*i+3)*z^2;
text(0,0.98-(i+1)*0.04,['f(x)_',num2str(i+1),' = ',num2str(C(3*i+1)), ' +
',num2str(C(3*i+2)), 'x + ',num2str(C(3*i+3)), 'x^2'])
end
axis off
% The following plots the splines as well as finding the maximum and minimum y value
% These values are needed for plotting
figure('Position',wind)
title('Quadratic spline interpolation','Fontweight','bold','FontSize',14)
for i=0:n-2
f=C(3*i+1)+C(3*i+2)*z+C(3*i+3)*z^2;
if xdesired < x(i+2) & xdesired >= x(i+1)
fxqdesired=subs(f,z,xdesired);
end
for j = x(i+1):(x(i+2)-x(i+1))/4):x(i+2)
if subs(f,z,j) > ymax
ymax = subs(f,z,j);
end
if subs(f,z,j) < ymin
ymin = subs(f,z,j);
end
end
ezplot(f,[x(i+1),x(i+2)])
hold on
end
text(0.1*(max(x)-min(x)),0.8*(max(y)-min(y)),['f(',num2str(xdesired),') =
',num2str(fxqdesired)])
plot(x,y,'ro','MarkerSize',10,'MarkerFaceColor',[1,0,0])
plot(xdesired,fxqdesired,'kx','Linewidth',2,'MarkerSize',12)
title('Quadratic spline interpolation','Fontweight','bold','FontSize',14)
xlim([min(x) max(x)])
tot=(ymax-ymin)*0.05;
ylim([ymin-tot ymax+tot])

```

Dari program di atas, maka grafik yang dihasilkan adalah sebagai berikut.



Grafik interpolasi kuadrat spline dengan Matlab

Persamaan yang dihasilkan dari interpolasi kuadrat spline di atas menjadi

QUADRATIC SPLINE INTERPOLATION	
$f(x_1)$	$= 7.2525 + 0.014125X + 0x^2$
$f(x_2)$	$= 5.4104 - 4.5e - 005x - 2.725e - 005x^2$
$f(x_3)$	$= 6.0888 + 0.00843x - 7.5e - 007x^2$
$f(x_4)$	$= 6 + 0.005475x - 2.5417e - 005x^2$
$f(x_5)$	$= 6 + 0.005475x + 5e - 006x^2$

Gambar 2. Output persamaan interpolasi linear dengan Matlab

### Daftar Pustaka

-----, 2006, *Spline Interpolation in Matlab*, diakses dari [www.math.colostate.edu/~gerhard/classes/331/lab/spline.pdf](http://www.math.colostate.edu/~gerhard/classes/331/lab/spline.pdf) pada tanggal 10 Februari 2006

-----, 2006, *Spline Toolbox*, diakses dari <http://www.eecs.umich.edu/dco/faq/matlab-6.5/help/toolbox/splines/preface3.html> pada tanggal 2 Februari 2006

-----, 2006, *Spline Interpolation*, diakses dari  
[http://en.wikipedia.org/wiki/Spline\\_interpolation](http://en.wikipedia.org/wiki/Spline_interpolation) pada tanggal 15 Februari  
2006

Collier, N., Kaw, A., 2006, *Spline Interpolation Simulation : Graphical  
Simulation of The Method Language : Matlab r12*, diakses dari  
[http://numericalmethods.eng.usf.edu/mtl/mec/05inp/  
mtl\\_mec\\_inp\\_sim\\_spline.pdf#search='quadratic%20spline%20interpolation  
%20matlab'](http://numericalmethods.eng.usf.edu/mtl/mec/05inp/mtl_mec_inp_sim_spline.pdf#search='quadratic%20spline%20interpolation%20matlab') pada tanggal 20 Februari 2006

Lasijo, 2001, Fitting Kurva Dengan Menggunakan Spline Kubik, *Integral*, vol. 6,  
no. 2, Oktober 2001